

# Java语言环境介绍

# 目录

---

◆ Java简介

◆ JDK

◆ Java程序语法结构

◆ 开发工具

- ▶ Java来自于Sun公司的一个叫Green的项目，其原先的目的是为家用消费电子产品开发一个分布式代码系统，这样我们可以把E-mail发给电冰箱、电视机等家用电器，对它们进行控制，和它们进行信息交流。开始，准备采用C++，但C++太复杂，安全性差，最后基于C++开发了一种新的语言Oak（Java的前身），Oak是一种用于网络的精巧而安全的语言，Sun公司曾以此投标一个交互式电视项目，但结果被SGI打败。

- ▶ 可怜的Oak几乎无家可归，恰巧这时Mark Ardreesen开发的Mosaic和Netscape启发了Oak项目组成员，他们用Java编制了HotJava浏览器，得到了Sun公司首席执行官Scott McNealy的支持，触发了Java进军Internet。

Java的取名也有一则趣闻，有一天，几位Java成员组的会员正在讨论给这个新的语言取什么名字，当时他们正在咖啡馆喝着Java(爪哇)咖啡，有一个人灵机一动说就叫Java怎样，得到了其他人的赞赏，于是，Java这个名字就这样传开了。



James Gosling领导的Green小组开发了面向数字家电的Oak语言

Internet迅猛发展Java语言正式问世被美国著名杂志PC Magazine 评为1995年十大优秀科技产品之一

推出了Java2平台。性能上有所提高，而且对Java的基本模式进行了完善，并提供了更为丰富的API

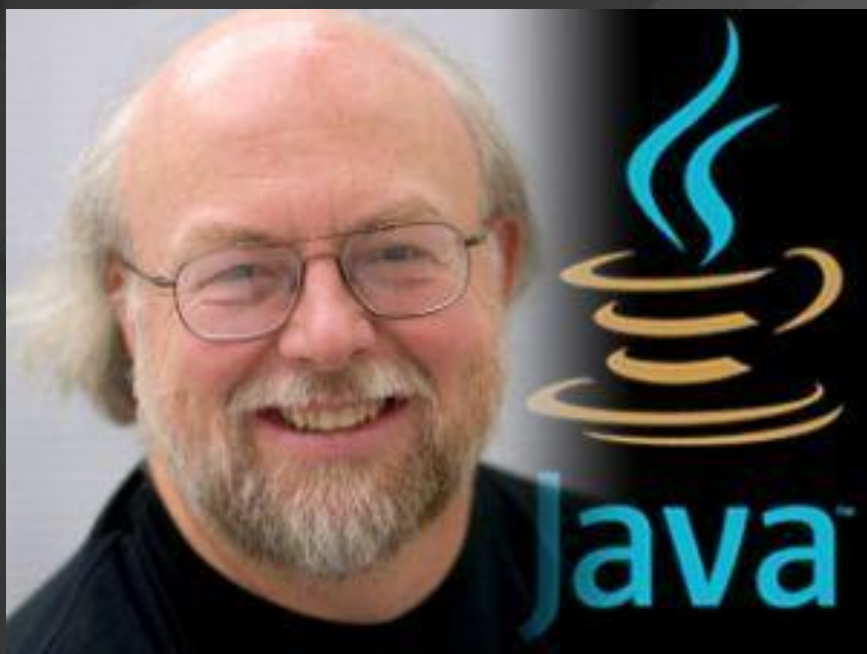
# Java之父James Gosling

- ▶ 詹姆斯·高斯林出生于加拿大，是一位计算机编程天才。在卡内基·梅隆大学攻读计算机博士学位时，他编写了多处理器版本的[Unix](#)操作系统。
- ▶ 1991年，在[Sun](#)公司工作期间，高斯林和一群技术人员创建了一个名为Oak的项目，旨在开发运行于虚拟机的编程语言，同时允许程序在电视机机顶盒等多平台上运行。后来，这项工作就演变为Java。随着[互联网](#)的普及，尤其是网景开发的网页[浏览器](#)的面世，Java成为全球最流行的开发语言。因此被人称作Java之父。





# ◆ Java之父 James Gosling



- ▶ 在1984年到2010年间，高斯林一直供职于Sun微系统公司。
- ▶ 在2010年甲骨文收购Sun后不久，这位Java编程语言的发明者宣布离职，并在2011年初加入谷歌。
- ▶ 2011年8月30日，高斯林在其博客上宣布离开谷歌，加入开展海洋探测业务的机器人制造公司Liquid Robotics，任首席软件架构师。负责传感器软件开发和自主导航设计，数据中心海量数据处理。

- ▶ Java是Sun Microsystems于1995年推出的高级编程语言
- ▶ Java 领域三个领域：
  - ▶ JavaSE，标准版（J2SE）  
java的标准版 包含了语法 基础 等很多基础的知识  
是必须要掌握的！



- ▶ JavaEE，企业版（J2EE）  
偏向于企业应用的 主要是SSH三大框架
- ▶ JavaME，微缩版（J2ME）  
手持设备程序的开发的部分

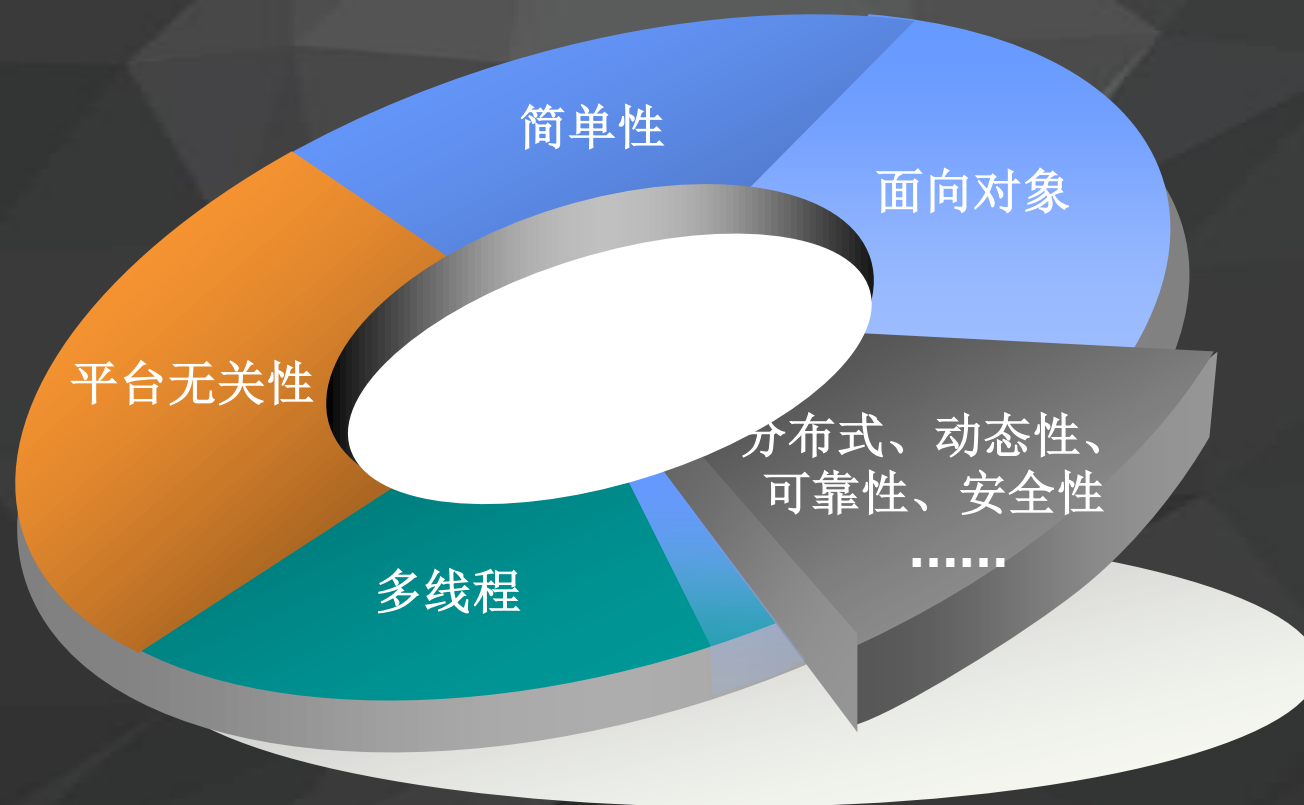


# Java可以做什么

- ▶ 开发桌面应用程序
  - ▶ 银行软件
  - ▶ 商场结算软件
- ▶ 开发面向Internet的应用程序
  - ▶ 网上数码商城
  - ▶ 阿里巴巴
  - ▶ 易趣网
- ▶ 手机、PDA、电视机顶盒、打印机
- ▶ Android应用程序



# 为什么要用Java





# 简单性

- ▶ Java最初是为对家用电器进行集成控制而设计的一种语言，因此它必须简单明了。Java语言的简单性主要体现在三个方面：
  - ▶ Java的风格类似于C++，因而C++程序员初次接触Java语言，就会感到很熟悉。从某种意义上讲，Java语言是C及C++语言的一个变种，因此，C++程序员可以很快地掌握Java编程技术。
  - ▶ Java摒弃了C++中容易引发程序错误的一些特性，如指针、结构、枚举以及内存管理等。
  - ▶ Java提供了丰富的类库，可以帮助我们很方便的开发Java程序。

- ▶ 面向对象可以说是Java最重要的特性，所以它支持继承、重载、多态等面向对象的特性。
- ▶ Java语言的设计是完全面向对象的,它不支持类似C语言那样的面向过程的程序设计技术。



# ◆ 健壮、可靠的

- ▶ Java致力于检查程序在编译和运行时的错误。
- ▶ Java也是一种强类型的语言，其类型检查比C++还要严格。类型检查帮助我们检查出许多开发早期出现的错误。
- ▶ Java自己负责内存管理，提供了垃圾内存回收机制，有效的避免了C++中最头疼的内存泄漏问题。







- ▶ **Java的安全性可从两个方面得到保证。**
  - ▶ 一方面，在Java语言里，删除了指针和释放内存等C++功能，避免了非法内存操作。
  - ▶ 另一方面，通过Java的安全体系架构来确保Java代码的安全性。当我们从网上下载Java代码在本地执行时，Java的安全架构能确保恶意的代码不能随意访问我们本地计算机的资源，例如：删除文件，访问本地网络资源等操作都是被禁止的。

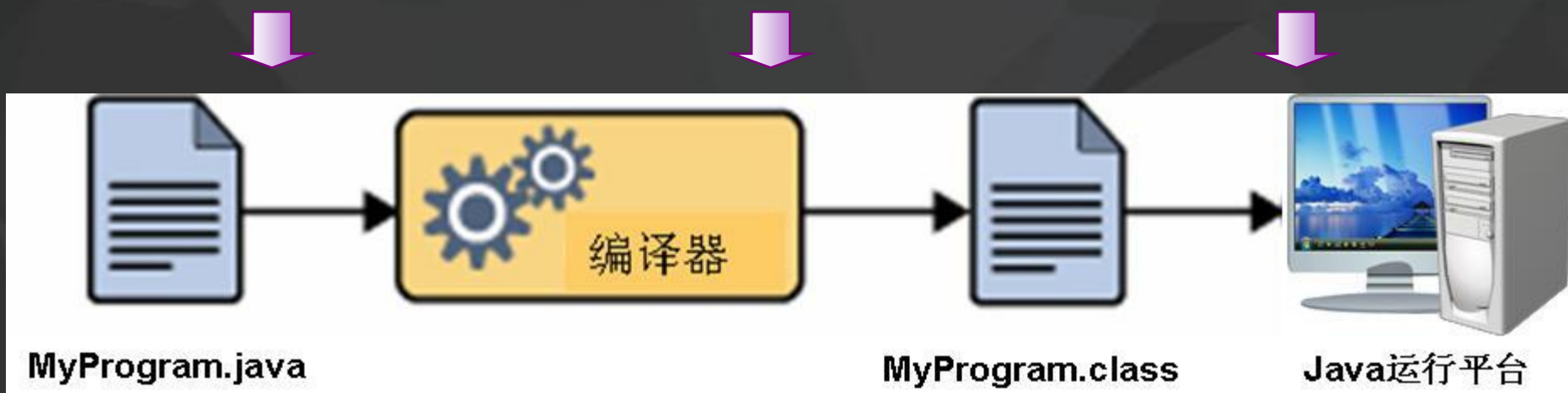
# 与平台无关的

- ▶ Java作为一种网络语言，其源代码被编译成一种结构中立的中间文件格式。只要有Java运行系统的机器都能执行这种中间代码。Java源程序被编译成一种与机器无关的字节码格式，在Java虚拟机上运行。

1、编写源程序

2、编译源程序

3、解释运行



# 多线程的

- ▶ Java语言的一个重要特性就是在语言级支持多线程的程序设计。
- ▶ 多线程就好像我们做一张桌子，如果你一个人完成这张桌子—就好像单线程，那么你需要先做桌面，做完桌面后，再做4个桌子腿，如果现在有5个人来做桌子—在程序中开辟5个线程，其中1个人做桌面，另外4个人分别做4个桌子腿，那么这两种方式效率的高低，相信大家都能区分出来。



- ▶ 简单的
- ▶ 面向对象的
- ▶ 动态、健壮、安全的
- ▶ 解释的
- ▶ 与平台无关的
- ▶ 多线程的

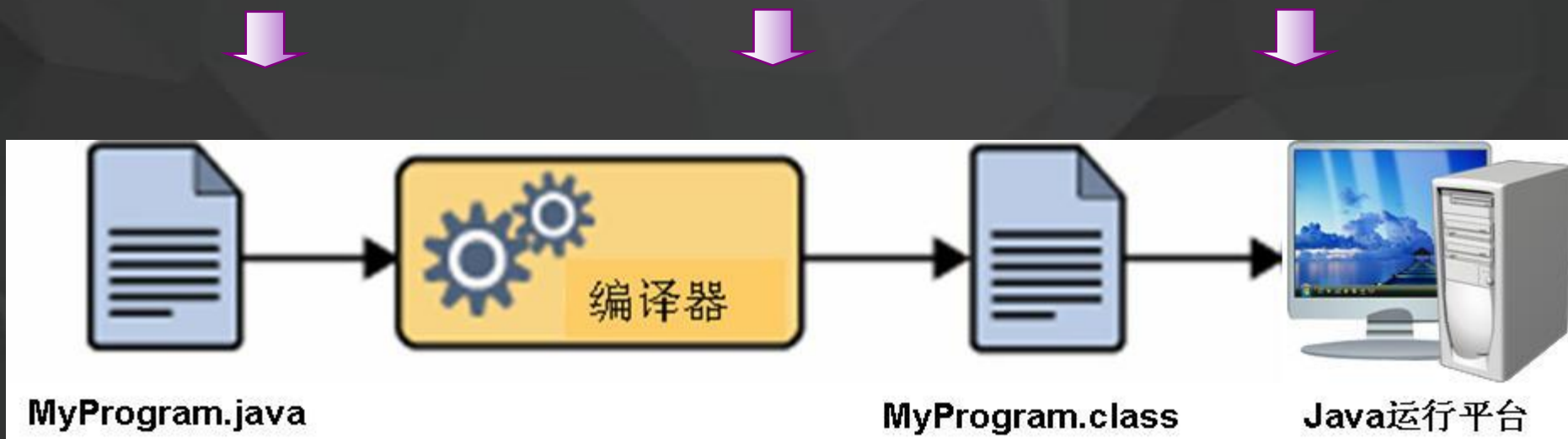
# 关于Java语言的跨平台

- ▶ Java的跨平台是通过Java虚拟机（JVM）来实现的。
- ▶ 所有的java程序会首先被编译为.class的类文件，这种类文件可以在虚拟机上执行

1、编写源程序

2、编译源程序

3、解释运行





- ▶ **JDK ( java development kit )**
  - ▶ JDK是java开发工具包
- ▶ **我们开发的实际情况是：**
  - ▶ 我们利用JDK ( 调用JAVA API ) 开发了属于我们自己的JAVA程序后
  - ▶ 通过JDK中的编译程序 ( javac ) 将我们的文本java文件编译成JAVA字节码
  - ▶ 在JRE上运行这些JAVA字节码，JVM解析这些字节码，映射到CPU指令集或OS的系统调用。

Java程序

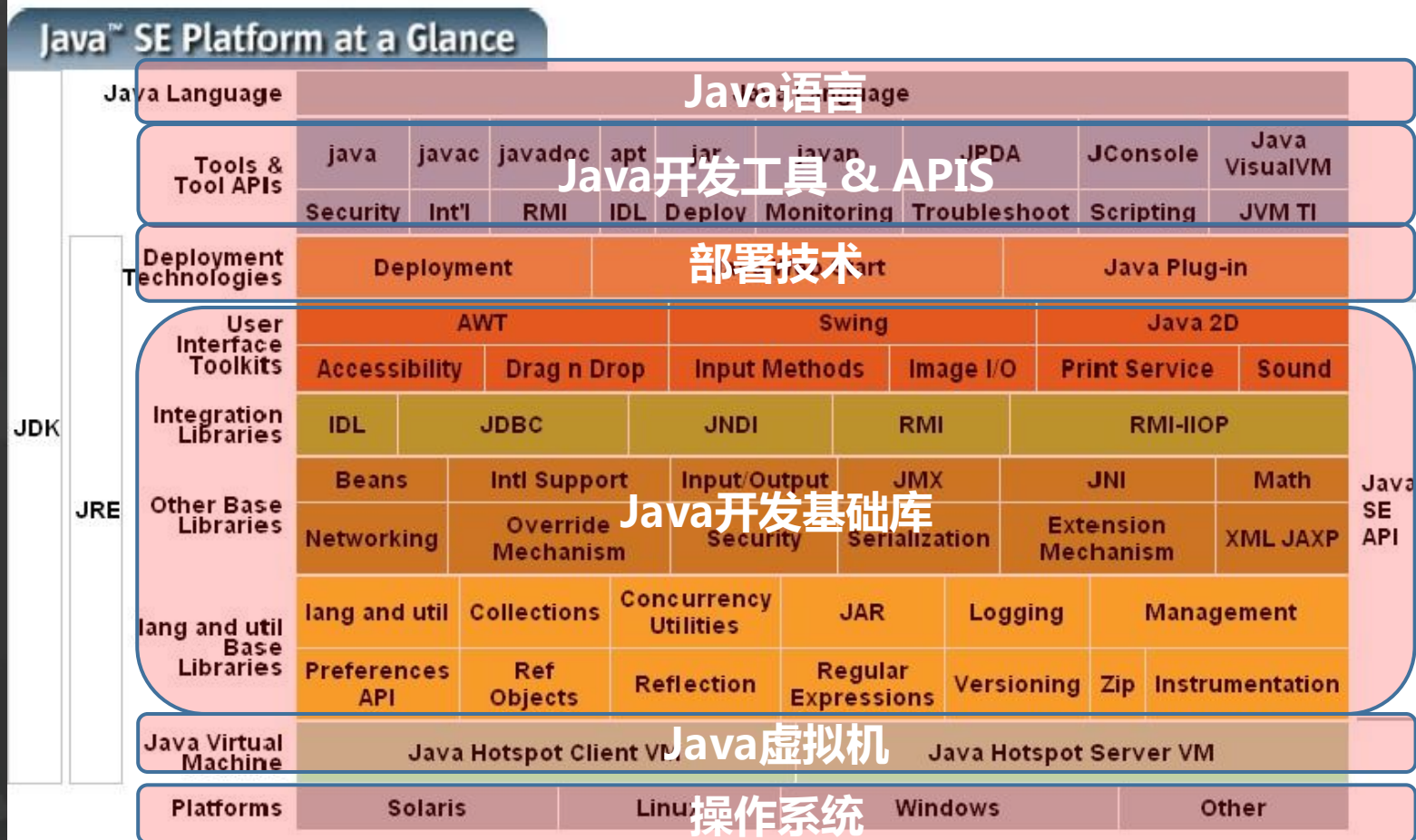
编译为class字节码

JVM解析字节码














# 关于JDK

## Java平台



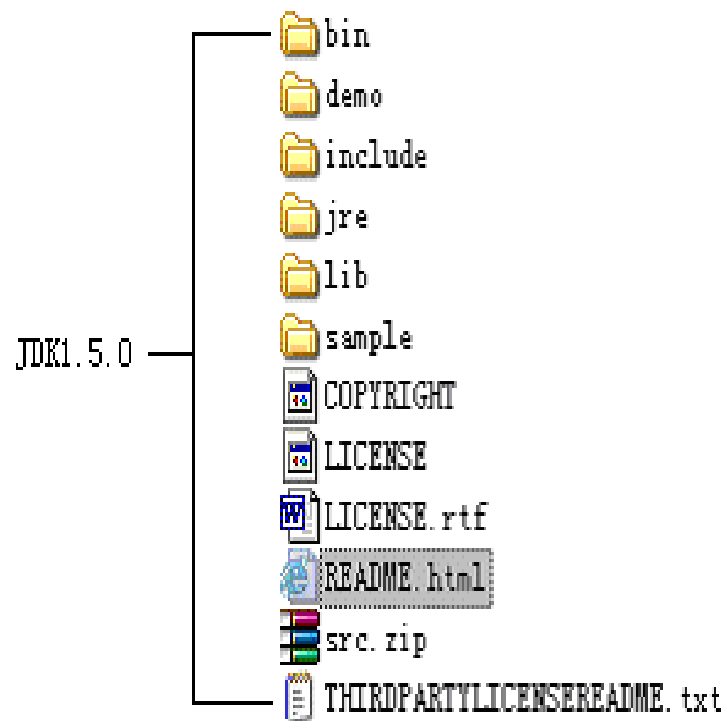
下载网址：

<http://www.oracle.com/technetwork/java/index.html>

Java SE Development Kit 8u20		
You must accept the <a href="#">Oracle Binary Code License Agreement for Java SE</a> to download this software.		
<input type="radio"/> Accept License Agreement <input checked="" type="radio"/> Decline License Agreement		
Product / File Description	File Size	Download
Linux x86	135.24 MB	 <a href="#">jdk-8u20-linux-i586.rpm</a>
Linux x86	154.87 MB	 <a href="#">jdk-8u20-linux-i586.tar.gz</a>
Linux x64	135.6 MB	 <a href="#">jdk-8u20-linux-x64.rpm</a>
Linux x64	153.42 MB	 <a href="#">jdk-8u20-linux-x64.tar.gz</a>
Mac OS X x64	209.11 MB	 <a href="#">jdk-8u20-macosx-x64.dmg</a>
Solaris SPARC 64-bit (SVR4 package)	137.02 MB	 <a href="#">jdk-8u20-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	97.09 MB	 <a href="#">jdk-8u20-solaris-sparcv9.tar.gz</a>
Solaris x64 (SVR4 package)	137.16 MB	 <a href="#">jdk-8u20-solaris-x64.tar.Z</a>
Solaris x64	94.22 MB	 <a href="#">jdk-8u20-solaris-x64.tar.gz</a>
Windows x86	161.08 MB	 <a href="#">jdk-8u20-windows-i586.exe</a>
Windows x64	173.08 MB	 <a href="#">jdk-8u20-windows-x64.exe</a>

# ◆ JDK目录结构

- ▶ bin目录中包含了所有JDK5.0提供的实用程序
- ▶ demo目录中包含了JDK5.0自带的实例程序
- ▶ include目录中包含了一些支持Java native方法的C\C++头文件
- ▶ jre目录中包含了Java运行环境所需的所有文件(JDK私有的)
- ▶ lib目录中包含了Java开发环境所需的库文件，它们以jar文件的形式保存
- ▶ sample目录中包含的是体现JDK5.0新特性的一些例程





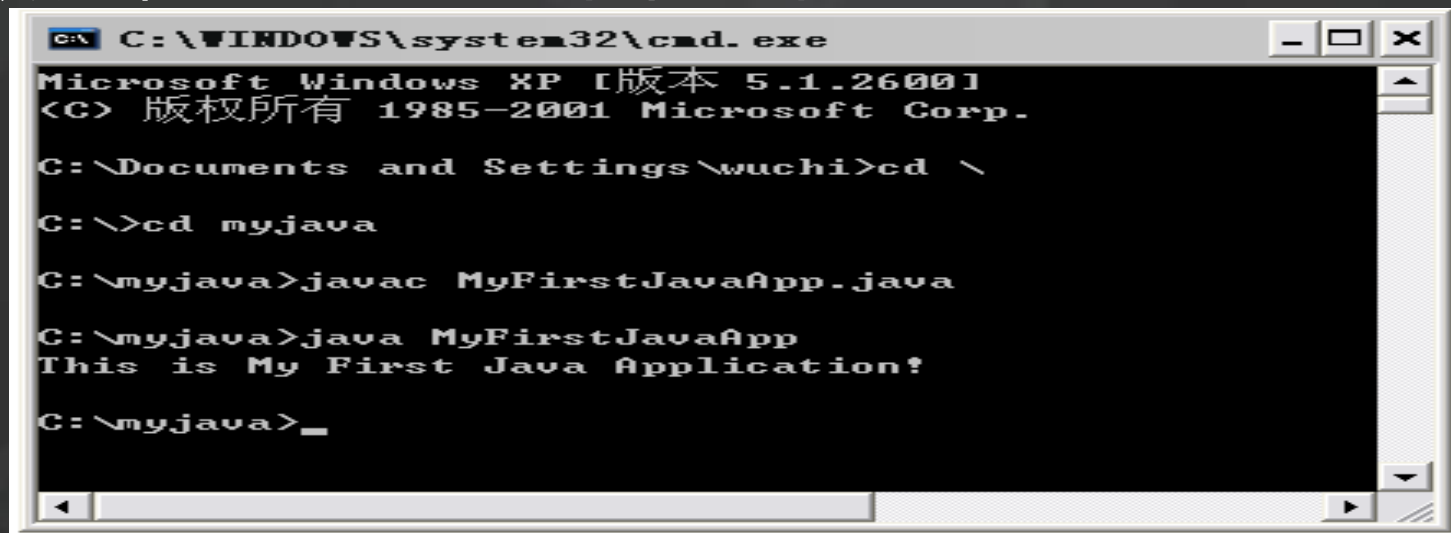
# JDK bin目录下实用程序

- ▶ **javac** : Java编译器，将Java源代码编译为字节码；
- ▶ **java** : Java解释器，用来解释执行Java程序的字节码文件；
- ▶ **javadoc** : 根据Java源代码及其说明语句生成的HTML文档；
- ▶ **appletviewer**(小程序浏览器)：一种执行HTML文件上的Java小程序类的Java浏览器；

```
public class MyFirstJavaApp {  
    public static void main(String[] args) {  
        System.out.println("This is My First Java Application!");  
    }  
}
```

# ◆ JDK bin目录下实用程序

- ▶ jdb : Java调试器，可以逐行地执行程序、设置断点和检查变量；
- ▶ javah : 产生可以调用Java过程的C过程，或建立能被Java程序调用的C过程的头文件；
- ▶ javap : Java反汇编器，显示编译类文件中的可访问功能和数据，同时显示字节代码含义。



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

C:\Documents and Settings\wuchi>cd \

C:\>cd myjava

C:\myjava>javac MyFirstJavaApp.java

C:\myjava>java MyFirstJavaApp
This is My First Java Application!

C:\myjava>_
```

# ◆ 关于环境变量的配置

- ▶ 开始 - 运行 - CMD - 输入javac
- ▶ 有且只有两种情况：
  - ▶ Javac成功运行
  - ▶ Javac不是可执行程序
- ▶ 环境变量的配置！
  - ▶ 当我们执行某个命令时，操作系统首先会在当前目录下查找该命令，如果没有找到这个命令程序，操作系统则会沿着path中所指定的目录依次查找，以最先找到的为准





# 关于环境变量的配置

- ▶ “我的电脑” - “属性” - “高级” - “环境变量” - “系统变量”
- ▶ 新建JAVA\_HOME变量，变量值：C:\Program Files\Java\jdk1.7.0
- ▶ 新建Path变量，变量值：%JAVA\_HOME%\bin;
- ▶ 通过运行cmd进入doc模式下，输入命令javac，检测是否配置成功
- ▶ 查看当前使用的JDK版本：java -version
- ▶ set + 变量名查看该变量的值



# 关于环境变量的配置

- ▶ **JAVA\_HOME就是安装目录.**
  - ▶ 为了方便引用, 比如, 你JDK安装在C:\Program Files\Java\jdk1.6.0目录里, 则设置JAVA\_HOME为该目录路径, 那么以后你要使用这个路径的时候, 只需输入%JAVA\_HOME%即可, 避免每次引用都输入很长的路径串;
  - ▶ 归一原则, 当你JDK路径被迫改变的时候, 你仅需更改JAVA\_HOME的变量值即可, 否则, 你就要更改任何用绝对路径引用JDK目录的文档, 要是万一你没有改全, 某个程序找不到JDK, 后果是可想而知的----系统崩溃!



# 关于环境变量的配置

---

- ▶ PATH:用于列出可执行文件的搜索路径.因为通过要通过命令行的命令启动JAVA,JAVAC等编译工具和解释工具,系统必须要知道他们所在的位置,PATH就是用来指明他们的位置的.



# 记事本开发Java程序

- ▶ 使用记事本编辑源程序，以.java为后缀名保存
- ▶ 使用javac命令编译.java文件，生成.class文件
- ▶ 使用java命令运行.class文件，输出程序结果

```
public class HelloWorld{  
    public static void main(String[] args){  
        System.out.println("Hello World!!!");  
    }  
}
```

命令提示符

```
E:\案例>java HelloWorld  
Hello World!!!
```

命令提示符

```
E:\案例>javac HelloWorld.java  
  
E:\案例>dir  
驱动器 E 中的卷没有标签。  
卷的序列号是 743F-2917  
  
E:\案例 的目录  
  
2010-03-11 14:41 <DIR> .  
2010-03-11 14:41 <DIR> ..  
2010-03-11 14:41 428 HelloWorld.class  
2010-03-11 14:17 115 HelloWorld.java  
2 个文件 544 字节  
2 个目录 81,904,902,144 可用字节
```

# Java程序语法结构



外层框架

Java入口程序  
框架

```
public class HelloWorld {  
  
    public static void main(String[] args) {  
        System.out.println("Hello World!!!");  
    }  
  
}
```

编写代码

# Java程序语法结构



关键字

类名与文件名完全一样

main方法四要素必不可少

main方法是Java程序执行的入口点

```
public class HelloWorld{
```

```
    public static void main(String[] args){
```

```
        System.out.println("Hello World!!!");
```

```
    }
```

```
}
```

从控制台输出信息

{和}一一对应，缺一不可





- **System.out.println ()**

```
public class HelloWorld{  
    public static void main(String[ ] args){  
        System.out.println("Hello World!!!");  
    }  
}
```

打印完引号中的信息后会自动换行

- **System.out.print()**

```
public class HelloWorld{  
    public static void main(String[ ] args){  
        System.out.print("Hello World!!!");  
    }  
}
```

打印输出信息后不会自动换行

问题



System.out.println()和System.out.print()有什么区别呢？

# 控制台输出



• 如何使System.out.println("");和System.out.print("\n"); 达到同样的效果？



○ 使用转义符

转义符	说 明
\n	将光标移动到下一行的第一格
\t	将光标移到下一个水平制表位置



```
public class HelloWorld{  
    public static void main(String[ ] args){  
        System.out.print("Hello World!!!\n");  
    }  
}
```



打印输出信息后  
将会自动换行



从控制台打印输出自己的信息

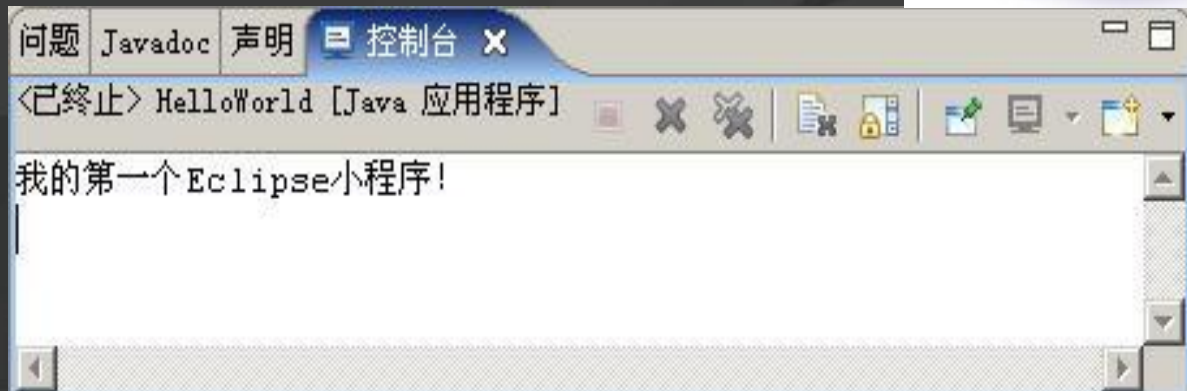
格式1：姓名  
          年龄

格式2：姓名    年龄

- ▶ 编码规范是作为一个程序员应该遵守的基本规则，是行业内大家都默守的做法
- ▶ 不遵守规范的代码不是好的代码，不是专业的代码
- ▶ Java编码规范
  - ▶ 一行只写一条语句
  - ▶ { }的使用及位置
  - ▶ 代码缩进

# 使用eclipse开发

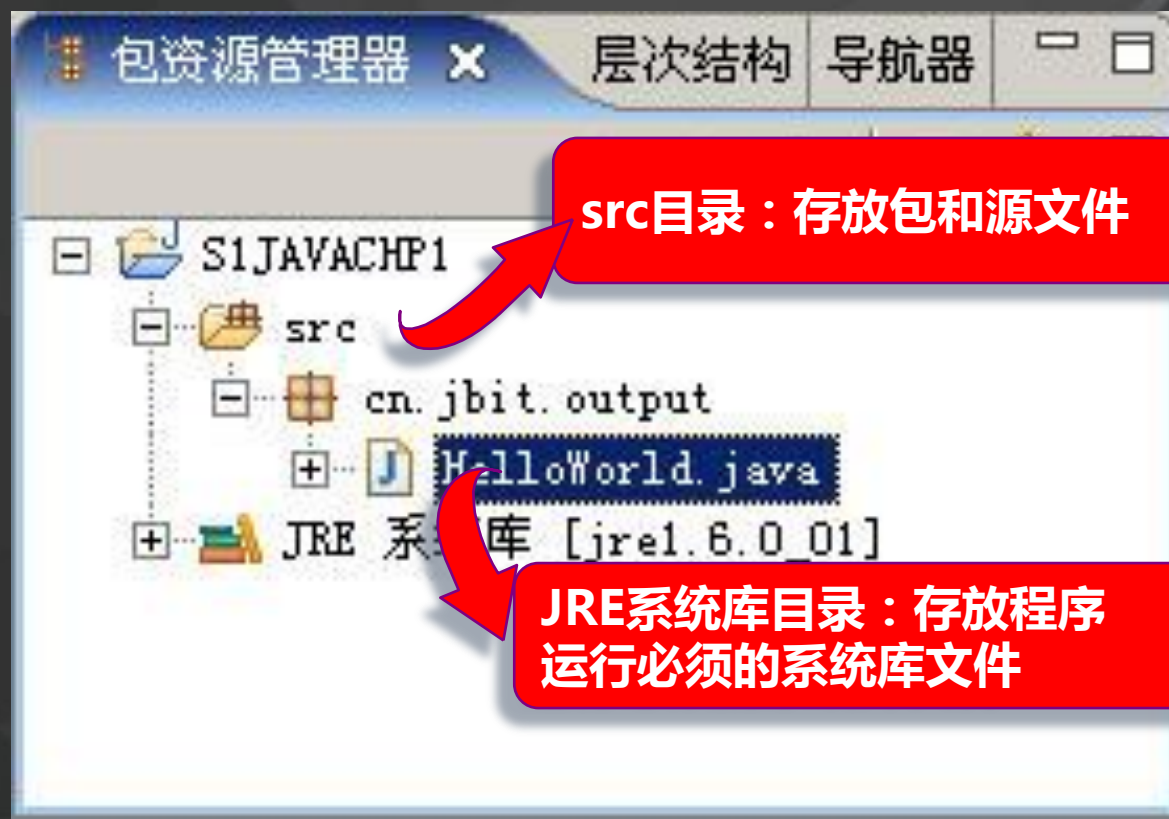
- ▶ 集成开发环境（IDE）是一类软件，它将程序开发环境和程序调试环境集合在一起，帮助程序员开发软件
- ▶ 使用Eclipse开发Java程序步骤：
  - ▶ 创建一个Java项目
  - ▶ 手动创建Java源程序
  - ▶ 编译和运行程序



# ◆包资源管理器

## ▶ 包资源管理器

- ▶ 用包组织Java源文件，类似于文件夹
- ▶ 选择菜单“窗口→显示视图→包资源管理器”打开







## • 需求说明：

- 使用Eclipse创建Java应用程序，实现从控制台输出多行信息：姓名、年龄、爱好

- ▶ java发展史
- ▶ java语言特性
- ▶ java跨平台原理
- ▶ SDK的下载安装及环境变量配置
- ▶ 第一个手写java程序
- ▶ 如何在doc下手动运行
- ▶ eclipse的下载安装及环境变量配置
- ▶ 第一个eclipse编写的java程序

# 谢 谢

